A new TCP/AQM for Stable Operation in Fast Networks

Fernando Paganini   Zhikui Wang   Steven H. Low   John C. Doyle

Abstract—**This paper is aimed at designing a congestion control system that scales gracefully with network capacity, providing high utilization, low queueing delay, dynamic stability, and fairness among users. In earlier work we had developed fluid-level control laws that achieve the first three objectives for arbitrary networks and delays, but were forced to constrain the resource allocation policy. In this paper we extend the theory to include dynamics at TCP sources, preserving the earlier features at fast time-scales, but permitting sources to match their steady-state preferences, provided a bound on round-trip-times is known.**

We develop two packet-level implementations of this protocol, using (i) ECN marking, and (ii) queueing delay, as means of communicating the congestion measure from links to sources. We discuss parameter choices and demonstrate using ns-2 simulations the stability of the protocol and its equilibrium features in terms of utilization, queueing and fairness. We also demonstrate the scalability of these features to increases in capacity, delay, and load, in comparison with other deployed and proposed protocols.

I. INTRODUCTION

Since their inception in the late 1980s [13], the congestion control mechanisms in TCP have been extremely successful in keeping the Internet under control while it underwent a dramatic growth. While improvements have been continually sought on issues such as retransmission, and on active queue management (AQM) [8], [6], [18], [10], the basic additive-increase-multiplicative-decrease (AIMD) structure in TCP congestion avoidance has remained unchanged. What is the incentive for research on replacing it?

One justification comes from the desire to improve the quality of service provided by the Internet, both in reducing queueing delays, and in allowing for more control over resource allocation, which is currently indirectly determined by the protocol. As argued in [12], [17], both objectives could be improved significantly by Explicit Congestion Notification (ECN) marks communicating *shadow prices* through the network, without the need for other higher complexity solutions that are being considered (e.g., differentiated services).

Another motivation comes from deficiencies of AIMD as the network further scales up in capacity. In such fast networks, congestion windows will easily scale up into the thousands, which creates two problems. On one hand, AI is too slow, since it can only change the window by one every round-trip-time, so changes of thousands can easily take minutes. On the other hand, MD is too fast: recent studies [7], [5], [21] have shown that in these high window regimes, TCP combined with RED [8] is unstable, leading to dramatic oscillations in network queues that not only cause delay jitter but can impact utilization.

It is quite possible that incremental modifications to the current protocols (e.g., ideas in [11]) could deal with these issues

in a satisfactory way, avoiding the deployment difficulties of a non-incremental solution. However, the above limitations (and others, such as deficiencies of a loss-based protocol in wireless networks) provide an incentive for another look at " doing it all again"; this is the motivation for this paper and other recent references (e.g., [15]).

We are further motivated by the large strides that have recently been taken in the analytical front, with tools from convex optimization coming into play to analyze resource allocation [16], [20], and advances in control theory to analyze stability [14], [23], [25], [19]; these, for the first time have been able to tackle the case of truly large-scale networks. In particular, the objectives of high utilization, low delay, fairness and stability appear to be within reach of protocols which imply a very modest modification of the current practice. We remark that although instabilities such as oscillations could perhaps be tolerated in the network context, the boundary of stability reflects the limits of predictable behavior in the network; the above control laws are aimed at operating precisely at that boundary.

Two alternative, mutually dual strategies have emerged: one "primal" approach [16], [14], [25] obtains stability and fairness by a new dynamics in TCP sources, combined with a static response from links; the link objectives (utilization, low queueing) are then pursued by a slower time-scale adaptation of AQM [19]. The "dual" approach of [20] and our earlier work [23] (reviewed in Section II), combines a dynamic AQM with a static TCP to achieve high utilization, low delay and dynamic stability for arbitrary networks and delays. What this solution does not allow is freedom in the resource allocation between TCP flows; instead, a response curve must be imposed on them, depending on their round-trip-time (RTT), that will determine their allocated throughput. A first contribution of this paper is to extend the theory in [23] to allow for an arbitrary choice of source utility functions; this could be used for instance to impose fairness among users who see the same bottleneck, independently of their RTT. This property is obtained by a new source control that uses separation of time-scales (dual to the strategy of [19]), running the "fairness" loop slower than a commonly agreed bound on the RTT. Other than this restriction, the stability proof extends to an arbitrary network.

The second objective of this paper is to go beyond fluid-flow models and pursue this family of protocols to the level of a packet implementation, within the constraints of mechanisms currently available in the Internet. Two strategies are pursued: one, described in Section IV, is based on the ECN bit to code the congestion information between links and sources; this version has the advantage of allowing operation with essentially zero delay, at the cost of some added complexity in network routers. Section V contains ns-2 simulation tests to demonstrate the performance of the protocol and comparisons with other versions

of TCP and AQM, in highly stressed congestion scenarios and high capacity links, and considering both long-lived flows and also file transfers drawn from a heavy-tailed distribution. The second implementation, described in Section VI, is based on queueing delay as a congestion measure, similar to what is done in TCP Vegas [2]. This allows some degradation of performance in terms of queueing delay and fairness, but has the advantage of requiring no explicit participation from routers. Preliminary simulation results are presented.

Conclusions are given in Section VII.

II. PROBLEM FORMULATION AND EARLIER WORK

A. Fluid-flow model and control objectives

We are concerned with a system of L communication links shared by a set of S sources. The routing matrix R, of dimensions $L \times S$, is defined by

$$R_{li} = \begin{cases} 1 & \text{if source } i \text{ uses link } l \\ 0 & \text{otherwise} \end{cases},$$

and assumed fixed. The theory will be based on a fluid-flow abstraction of the TCP/AQM congestion control problem. Each source i has an associated transmission rate $x_i(t)$; the set of transmission rates determines the aggregate flow $y_l(t)$ at each link, by the equation

$$y_l(t) = \sum_i R_{li} x_i(t - \tau_{li}^f), \tag{1}$$

in which the forward transmission delays τ_{li}^f between sources and links are accounted for. Each link has a capacity c_l in packets per second.

Next, we model the feedback mechanism which communicates to sources the congestion information about the network. The key idea is to associate with each link l a congestion measure or *price* $p_l(t)$ [16], [20], and assume sources have access to the *aggregate* price of all links in their route,

$$q_i(t) = \sum_l R_{li} p_l(t - \tau_{li}^b). \tag{2}$$

Here again we allow for backward delays τ_{li}^b in the feedback path from links to sources. As discussed in [22], this feedback model includes, to a good approximation, the mechanism present in existing protocols, with a different interpretation for price in different protocols (e.g. loss probability in TCP Reno, queueing delay in TCP Vegas). The total RTT for the source thus satisfies

$$\tau_i = \tau_{li}^f + \tau_{li}^b \tag{3}$$

for every link in the source's path. The vectors x, y, p, q collect the above quantities across sources and links.

In this framework, a congestion control system is specified by choosing (i) how the links fix their prices based on link utilization; (ii) how the sources fix their rates based on their aggregate price.

We remark that we are directly modelling only persistent sources, i.e. those long enough to be controlled. From the point of view of these "elephants", what matters mainly is that the system reaches an equilibrium point x_0, y_0, p_0, q_0 with high network utilization and adequate resource allocation among them.

The network is, however, also shared by short "mice", which don't last long enough to be controlled, and for which no "equilibrium" exists, but who are affected by the properties of the control, mainly through the queuing delay they experience. We will not model them explicitly here (they could be treated as noise in link rates), but will bear this objective in mind for our design.

Specifically, we lay out the following design objectives:
1. Network utilization. Link equilibrium rates y_{0l} should of course not exceed the capacity c_l, but also should attempt to track it.
2. Equilibrium queues should be empty to avoid delays.
3. Resource allocation. We will assume sources have a demand curve

$$x_{0i} = f_i(q_{0i}) \tag{4}$$

that specifies their desired equilibrium rate as a decreasing function of price. This is equivalent to assigning them a concave *utility function* $U_i(x_i)$, in the language of [16]; in this case $f_i = (U_i')^{-1}$. We would like the control system to reach an equilibrium that accommodates these demands. This does not in itself ensure fairness, or address differentiated services, but provides a tuning knob in which to tackle these kind of issues.
4. Stability. The equilibrium should be (at least locally) stable.

What makes this a challenging problem is that we require the above to hold for *arbitrary* networks, and that one must work with very tight information constraints: sources and links have only access to their respective variables, and nobody knows what the overall network is.

B. Control laws with scalable stability

We describe here the control laws of [23], which achieve three of the above objectives.

At the links, the price dynamics is defined as

$$\dot{p}_l = \begin{cases} \frac{y_l - c_{0l}}{c_{0l}}, & \text{if } p_l > 0 \text{ or } y_l > c_{0l}; \\ 0 & \text{otherwise,} \end{cases} \tag{5}$$

where c_{0l} is a target "virtual" capacity. At equilibrium, bottlenecks with nonzero price will have $y_{l0} = c_{0l}$, and non-bottlenecks with $y_{l0} < c_{0l}$ will have zero price. Note that with this choice the price is the "virtual" queueing delay one would get if the capacity was c_{0l}. Choosing c_{0l} slightly below the actual capacity c_l ensures high utilization and at the same time that the real queues are empty, as intended by the first two design objectives.

At the sources, a static rate function of the aggregate price is proposed:

$$x_i = x_{\max,i} \, e^{-\frac{\alpha_i q_i}{M_i \tau_i}}. \tag{6}$$

Here τ_i is the RTT, α_i a constant, and M_i a bound on the number of bottlenecks in source i's path. $x_{\max,i}$ is a maximum rate parameter, which can depend on M_i, τ_i (but not on q_i).

The above control laws are a special case of those in [20], and therefore define a unique equilibrium point. The main result of [23] is that the equilibrium is locally stable for arbitrary networks, parameters, and delays. This is shown by considering a perturbation $x = x_0 + \delta x, y = y_0 + \delta y, p = p_0 + \delta p, q = q_0 + \delta q$

around equilibrium, and writing the linearized equations in the Laplace domain:

$$\delta \bar{y}(s) = \bar{R}_f(s) \delta x(s), \qquad (7)$$

$$\delta q(s) = \bar{R}_b(s)^T \delta \bar{p}(s), \qquad (8)$$

$$\delta \bar{p} = \mathcal{C} \frac{I}{s} \delta \bar{y}, \qquad (9)$$

$$\delta x = -\mathcal{K} \delta q. \qquad (10)$$

Here we use the notation $\delta \bar{p}, \delta \bar{y}$ to indicate the reduced vectors obtained by eliminating non-bottleneck links, which do not contribute to the linear dynamics. Thus (1-2) linearize to (7-8), where the matrices $\bar{R}_f(s)$ and $\bar{R}_b(s)$ are obtained by eliminating non-bottleneck rows from R, and also replacing the "1" elements respectively by the delay terms $e^{-\tau_{i,l}^f s}$, $e^{-\tau_{i,l}^b s}$. The diagonal matrices

$$\mathcal{C} = \mathrm{diag}(\frac{1}{c_{0l}}), \quad \mathcal{K} = \mathrm{diag}(\kappa_i), \quad \kappa_i = \frac{\alpha_i x_{0i}}{M_i \tau_i}. \qquad (11)$$

are derived from the linearization of (5-6).

The above equations lead to a formula for the overall multi-variable loop transfer function

$$L(s) = \bar{R}_f(s) \mathcal{K} \bar{R}_b^T(s) \mathcal{C} \frac{I}{s}. \qquad (12)$$

The stability of such loops with integral control is studied in [23] via the following proposition.

Proposition 1: Consider a unity feedback loop, with $L(s) = F(s) \frac{I}{s}$. Suppose: (i) $F(s)$ is analytic in $Re(s) > 0$ and bounded in $Re(s) \geq 0$; (ii) $F(0)$ has strictly positive eigenvalues. (iii) For all $\gamma \in (0,1]$ and $\omega \neq 0$, the point -1 is not an eigenvalue of $\gamma L(j\omega)$. Then the closed loop is stable.

We have the following general result on scalable stability for arbitrary networks under a mild rank restriction.

Theorem 2 ([23]) Suppose the matrix $\bar{R} := \bar{R}_f(0) = \bar{R}_b(0)$ is of full row rank, and that $\alpha_i < \frac{\pi}{2}$. Then for arbitrary delays and link capacities, the unique equilibrium point of the system under control laws (5-6) is locally stable.

Taking $F(s) = \bar{R}_f(s) \mathcal{K} \bar{R}_b^T(s) \mathcal{C}$, it is easy to establish here that (i) holds, and (ii) follows from the rank assumption on $\bar{R}$. The more delicate step is (iii), as we will see below when discussing a generalization.

The laws of [23] satisfy the equilibrium objectives on the link side, and stability. However the exponential laws in (6) specify a fixed demand curve for the sources, or equivalently a fixed utility function. Some degrees of freedom are left in the choice of $x_{\max,i}$, and one could further generalize these laws as indicated in [23]. Nevertheless we do not have complete freedom in the choice of the demand curve, as we had aimed for in Section II-A. In particular, we would like to be able to eliminate the dependence of the equilibrium structure on the RTT, which is also present in current protocols. We remark that parallel work in [25] has derived solutions with scalable stability and arbitrary utility functions, but where the link utilization requirement is relaxed. Indeed, it appears that one must choose between the equilibrium conditions on either the source or the link side, if one desires a scalable stability theorem. In the next section we show how this difficulty is overcome if we slightly relax our scalability requirement.

III. A NEW CONTROL WITH ENHANCED FAIRNESS

The reason we are getting restrictions on source utility is that for static laws, the elasticity of the demand curve (the control gain at DC) coincides with the high frequency gain, and is thus constrained by stability. One way of decoupling the two gains is to replace the linearized source control by a dynamic, lead-lag compensation of the form

$$\delta x_i = -\frac{\kappa_i(s+z)}{s + \frac{z\kappa_i}{\nu_i}} \delta q_i. \qquad (13)$$

Here the high frequency gain κ_i is the same as in (11), "socially acceptable" from a dynamic perspective. The DC gain $\nu_i = -f_i'(q_{i0})$ is the elasticity of source demand based on its own "selfish" demand curve $x_{i0} = f_i(q_{i0})$, that need no longer be of the form (6). The *zero* z is assumed fixed across sources, a factor that will be essential to the stability proof below.

• If $\nu_i \leq \kappa_i$, a static source controller based on its utility would be within the limits of the earlier stability theorem, without any need for compensation. In this case, the above controller provides a *higher* gain at cross-over frequency, so that the network utilization loop reacts as fast as possible compatible with stability. It also gives phase *lead*, which reinforces the idea that stability is not compromised.

• If $\nu_i > \kappa_i$, the compensation forces the aggressive source to reduce its gain at cross-over frequency to maintain stability. Note that here the source pole is lower than z; the more aggressive the source tries to be, the slower this response becomes, to keep the high frequency behavior roughly intact. We also have a phase *lag* in this case, which means that care must be taken in the stability analysis.

A. Local Stability Results

With the new local source control, we will proceed to study the linearized stability of the closed loop, generalizing the method of Theorem 2. We first write down the overall loop transfer function

$$L(s) = R_f(s) \mathcal{K}(s) R_b^T(s) \mathcal{C} \frac{I}{s}, ha$$

which is analogous to (12) except that now

$$\mathcal{K}(s) = \mathrm{diag}(\kappa_i V_i(s)), \quad \text{with} \quad V_i(s) = \frac{s+z}{s + \frac{z\kappa_i}{\nu_i}},$$

κ_i as in (11). The stability argument is based again on Proposition 1, the main step being once more the study of the eigenvalues of $\gamma L(j\omega)$. As in [23], the key structure that is employed is the relationship

$$\bar{R}_b(s) = \bar{R}_f(-s) \mathrm{diag}(e^{-\tau_i s}),$$

which follows from (3), and allows us to write

$$L(j\omega) = R_f(j\omega) X_0 \mathcal{M} \Lambda(j\omega) R_f(j\omega)^* \mathcal{C},$$

$$\text{with} \quad X_0 = \mathrm{diag}(x_{0i}), \quad \mathcal{M} = \mathrm{diag}(\frac{1}{M_i}),$$

$$\Lambda(j\omega) = \mathrm{diag}(\lambda_i(j\omega)).$$

The only change with respect to [23] is that we have added the lead-lag term $V_i(s)$ to the diagonal elements of $\Lambda(s)$,

$$\lambda_i(s) = \frac{\alpha_i e^{-\tau_i s}}{\tau_i s} V_i(s). \tag{14}$$

Proceeding with the method of [23], we write $eig(\gamma L(j\omega)) = \mathrm{eig}\Big(\gamma P(j\omega)\Lambda(j\omega)\Big)$, where

$$P(j\omega) := \mathcal{M}^{\frac{1}{2}} X_0^{\frac{1}{2}} R_f(j\omega)^* \, \mathcal{C} \, R_f(j\omega) X_0^{\frac{1}{2}} \mathcal{M}^{\frac{1}{2}} \geq 0;$$

it follows in a similar way that $\rho(\gamma P) \leq \rho(P) \leq 1$. Then using Vinnicombe's lemma [24], the eigenvalues of $L(j\omega)$ are convex combinations of the $\lambda_i(j\omega)$, and the origin.

It remains to give conditions so that the convex combinations of the $\lambda_i(j\omega)$, which now include an extra lead-lag term, do not reach the critical point -1. Figure 1 contains various Nyquist plots of $\lambda_i(j\omega)$, for τ_i ranging between 1ms and 1sec, and ratios ν_i/κ_i ranging between 0.1 and 1000. The value of z is fixed at 0.2, and $\alpha = 1$.
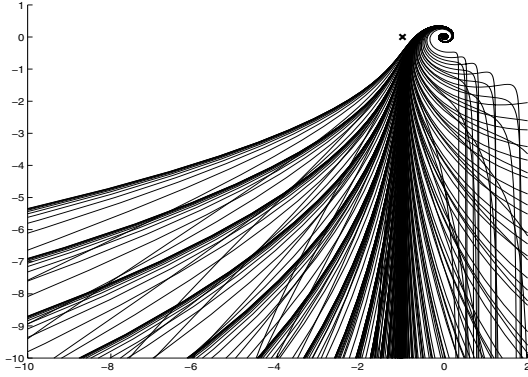

Fig. 1. Nyquist plots of $\lambda_i(j\omega)$, $z = 0.2$, $\alpha = 1$, various τ_i and ν_i/κ_i.

A first comment is that here the plots do not coincide, as they did in the "scale-invariant" case of [23], when we used the source control κ_i (only the high frequency portion of the above plots coincide).

Secondly, we note that there is not an obvious separation between the convex hull of these points and the critical point -1. One could think of obtaining convex separation through a slanted line; this however, would imply a lower limit $-\pi + \theta$, $\theta > 0$ on the phase of $\lambda_i(j\omega)$ at low frequencies, which in turn implies, based on (14), a limit on the lag-lead gain ratio ν_i/κ_i. This may be acceptable, but would not allow us to accommodate *arbitrary* utilities.

The alternative is to treat the low-frequency portion of the above curve separately, ensuring for instance that it doesn't reach phase $-\pi$. This, however, implies a common notion of what "low-frequency" means, so that we are not operating in different portions of the curve for sources with different RTTs. This can be obtained through a fixed bound $\bar{\tau}$ on the RTT, as follows.

Proposition 3: Assume that for every source i, $\tau_i \leq \bar{\tau}$. In the source controllers (13), choose $\alpha_i = \alpha < \frac{\pi}{2}$ and $z = \frac{\eta}{\bar{\tau}}$. Then for a small enough $\eta \in (0, 1)$ depending only on α, $-1 \notin eig(L(j\omega))$.

Fig. 2. Plots of ∇ (solid) and ∇_η (dashed)
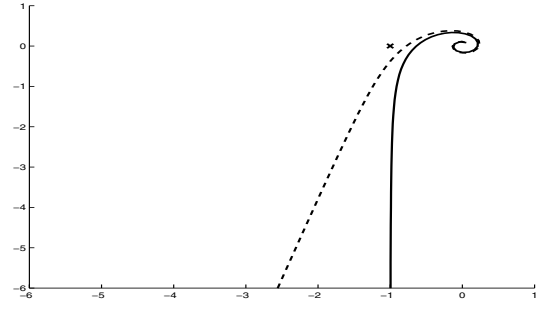
Proof:

For fixed $\alpha < \frac{\pi}{2}$, let $\nabla = \{\frac{\alpha e^{-j\theta}}{j\theta} : \theta > 0\}$. This curve, depicted by the solid curve in Figure 2, would be the Nyquist plot of $\lambda_i(j\omega)$ for $\nu_i = \kappa_i$. ∇ is strictly away from the critical point.

We now quantify the extra gain and phase introduced by the lag-lead $V_i(j\omega)$ for frequencies $\omega \geq \frac{1}{\bar{\tau}}$:

$$\left| \frac{j\omega + z}{j\omega + p} \right| \leq \sqrt{1 + \frac{z^2}{\omega^2}} \leq \sqrt{1 + \eta^2},$$

$$\mathrm{phase}\left(\frac{j\omega + z}{j\omega + p} \right) \geq -\arctan(\frac{z}{\omega}) \geq -\arctan(\eta)$$

This means for this frequency range, $\lambda_i(j\omega)$ will lie always below the perturbed curve

$$\nabla_\eta := \sqrt{1 + \eta^2} e^{-j \arctan(\eta)} \nabla.$$

(a slight clockwise rotation and expansion of ∇), depicted by dashed lines in Figure 2. By appropriately small choice of η, depending only on α, we can make sure that this curve stays below the critical point. It follows that convex combinations of $\lambda_i(j\omega)$ cannot reach the critical point for $\omega \geq \frac{1}{\bar{\tau}}$.

It remains to consider the frequencies $\omega \in (0, \frac{1}{\bar{\tau}})$. We will argue that in this frequency range, $\lambda_i(j\omega)$ is always in the lower half-plane (negative imaginary part), and hence again one cannot obtain the critical point by convex combinations.

To see this, compute

$$\mathrm{phase}(\lambda_i(j\omega)) = -\frac{\pi}{2} - \tau_i\omega + \mathrm{phase}(V_i(j\omega))$$

$$> -\pi - \tau_i\omega + \arctan(\frac{\omega}{z})$$

$$\geq -\pi - \bar{\tau}\omega + \arctan(\frac{\bar{\tau}\omega}{\eta})$$

Thus it suffices to show that for $\omega \in (0, \frac{1}{\bar{\tau}})$,

$$\arctan(\frac{\bar{\tau}\omega}{\eta}) > \bar{\tau}\omega,$$

or equivalently

$$\eta < \frac{\bar{\tau}\omega}{\tan(\bar{\tau}\omega)}$$

The right hand-side is decreasing in $\bar{\tau}\omega < 1$, so it suffices to choose $\eta < \frac{1}{\tan(1)} \approx 0.64$. ∎

B. Nonlinear implementation

We now discuss how to embed our new linearized source control law in global nonlinear laws. The requirements are:

• The equilibrium matches the desired utility function, $U_i'(x_{0i}) = q_{0i}$, or equivalently the demand curve (4) for $f_i = (U_i)^{-1}$.

• The linearization is (13), with the zero z being fixed, independently of the operating point and the RTT.

We now present a nonlinear implementation that satisfies these conditions, of a similar nature to laws obtained in the "primal" approach [16], [14], [25].

$$\tau_i \dot{\xi}_i = \beta_i(U_i'(x_i) - q_i), \qquad (15)$$

$$x_i = x_{m,i} e^{\left(\xi_i - \frac{\alpha_i q_i}{M_i \tau_i}\right)}. \qquad (16)$$

Note that (16) corresponds exactly to the rate control law in (6), with the change that the parameter $x_{\max}$ is now varied exponentially as $x_{\max,i} = x_{m,i} e^{\xi_i}$, with ξ_i as in (15). If β_i is small, the intuition is that the sources use (6) at fast time-scales, but slowly adapt their $x_{\max_i}$ to achieve an equilibrium rate that matches their utility function, as follows clearly from equation (15).

Remark: The RTT used in (15-16) could be the real-time measurement, or instead, it could be replaced by the fixed portion d_i (propagation/processing delay). Both will coincide locally around an equilibrium with empty queues, but the latter option is generally preferable because it avoids a more complex time-varying dynamics during a transient where there are queueing delays.

We now find the linearization around equilibrium; the source subscript i is omitted for brevity. For increments $\xi = \xi_0 + \delta\xi$, $x = x_0 + \delta x$, $q = q_0 + \delta q$, we obtain the linearized equations:

$$\tau\delta\dot{\xi} = \beta(U''(x_0)\delta x - \delta q) = \beta\left(-\frac{\delta x}{\nu} - \delta q\right), \qquad (17)$$

$$\delta x = x_0\left(\delta\xi - \frac{\alpha}{M\tau}\delta q\right) = x_0\delta\xi - \kappa\delta q. \qquad (18)$$

Here we have used the fact $U''(x_0) = \frac{1}{f'(q_0)} = -\frac{1}{\nu}$, and the expression (10) for κ. Some algebra in the Laplace domain leads to the transfer function

$$\delta x = -\kappa \left(\frac{s + \frac{\beta x_0}{\kappa\tau}}{s + \frac{\beta x_0}{\nu\tau}} \right) \delta q,$$

that is exactly of the form in (13) if we take

$$z = \frac{\beta x_0}{\kappa\tau} = \frac{\beta M}{\alpha}.$$

By choosing β, the zero of our lead-lag can be made independent of the operating point, or the delay, as desired.

We recapitulate the main result as follows.

Theorem 4: Consider the source control (15-16) where $U_i(x_i)$ is the source utility function, and the link control (5). At equilibrium, this system will satisfy the desired demand curve $x_{i0} = f_i(q_{i0})$, and the bottleneck links will satisfy $y_{0l} = c_{0l}$, with empty queues. Furthermore, under the rank assumption in Theorem 2, $\alpha_i < \frac{\pi}{2}$, and $z = \frac{\beta_i M_i}{\alpha_i}$ chosen as in Proposition 3, the equilibrium point will be locally stable.

We have thus satisfied all the objectives set forth in Section II-A, except for the fact that an overall bound on the RTT had to be imposed.

IV. A PACKET-LEVEL IMPLEMENTATION USING ECN

So far we have worked with the abstraction of the congestion control problem laid out in Section II-A. In this section we transition these ideas to an actual packet-level protocol, as could be implemented in a real world network.

A first comment is that while we have assumed that source can control *rates*, in practice they adapt their congestion window w_i; its effect over the rate can be approximately described, over time-scales longer than the RTT, by the relationship

$$x_i \approx \frac{w_i}{\tau_i}.$$

Sources should set w_i so that the rate targets the desired "equation-based" value from (15-16), with a suitable time discretization. To make the discussion concrete, in this section we use the utility function $U_i(x_i) = K_i \log(x_i)$, which induces the so-called "proportional fairness" [16].

A. Marking and Estimation

The key issue for the implementation of the above protocols is the communication of price signals from links back to sources, in an additive way across the source's route. In this section we explore the use of an Explicit Congestion Notification (ECN) bit to implement this feature. A natural way to obtain the additivity property is by Random Exponential Marking (REM, [1]), in which an ECN bit would be marked at each link l with probability $1 - \phi^{-p_l}$, where $\phi > 1$ is a global constant. Assuming independence between links, the overall probability that a packet from source i gets marked is (see [1])

$$\mathcal{P}_i = 1 - \phi^{-q_i},$$

and therefore q_i can be estimated from marking statistics. For example, a shift-register of the last N received marks can be maintained, the fraction of positive marks providing an estimate $\hat{\mathcal{P}}_i$ of the marking probability, from which an estimate $\hat{q}_i$ can be derived, and used in place of q_i in the source equations.

While simple in principle, two related issues are important to make this scheme practical:

1. The choice of a universal ϕ across the network means choosing a scale of prices for which our estimation will be most accurate (i.e., where the marking probability is not too close to 0 or 1). For instance, for $\phi = 1000$ the range of prices (in seconds) $[0.007, 0.43]$ corresponds to marking probabilities between 5% and 95%. Source demand functions (4) should be tailored to operate in this range. Below, this will be taken into account the choice of the parameters of our utility function.

2. An estimation based on a moving average of size N introduces an additional *delay* in the feedback loop, of approximately

$$\tau_{est} \approx \frac{N}{2w}\tau, \qquad (19)$$

which is the time it takes to receive $\frac{N}{2}$ packets. This delay could compromise stability, a factor that can partly be addressed by

choosing α away from the stability limit. Still, it is clear from (19) that one should avoid high estimation windows, so there is compromise between stability and accurate price estimation.

B. Discretized source and link laws

The next step is to describe a discrete algorithm that approximates the continuous-time laws we have described. The following equations provide a congestion window update, with discretization interval T_s, consistent with the chosen utility function:

$$\xi_i(k) = \xi_i(k-1) + \beta_i \left(\frac{K_i}{w_i(k-1)} - \frac{q_i(k)}{d_i} \right) T_s, \quad (20)$$

$$w_i(k) = w_{m,i} e^{\left(\xi_i(k) - \frac{\alpha_i q_i(k)}{M_i d_i} \right)}. \quad (21)$$

Note that we have explicitly replaced the RTT by the fixed portion d_i, consistently with the remark in the previous section. In practice, this quantity can be estimated through the minimum observed RTT during the TCP session (termed $baseRTT$ below). The source operations are described by the pseudo-code in Figure 3. The price from the links is estimated from the ECN bits in the latest N packets on every ACK arrival. The difference equations (20-21) are used to calculate the expected congestion window. Since estimation noise will feed through to the congestion window by (21), we impose caps on the change of the congestion window per ACK to mitigate it. Furthermore, the output packet flows are paced uniformly over each RTT.

Every $intInterval$ seconds:

$$\xi \leftarrow \xi + \beta * \left(\frac{K}{expWnd} - \frac{estPrice}{baseRTT} \right) * intInterval;$$

$$expWnd \leftarrow W_m * exp \left(\xi - \frac{\alpha * estPrice}{M * baseRTT} \right);$$

On each ACK arrival:

    Estimate $estProb$ using last N ACKs;
    $tmp \leftarrow CWnd - expWnd;$
    $if(tmp > maxDecrement)$
        $CWnd \leftarrow CWnd - maxDecrement;$
    $elseif(tmp < -maxIncrement)$
    $CWnd \leftarrow CWnd + maxIncrement;$
    $else$
        $CWnd \leftarrow expWnd;$

Variables:
ξ: state variable;
$estPrice$: estimated price from the marking probability;
$expWnd$: expected congestion window.

Parameters:
α: stability parameter;
β: constant from the zero point;
Φ: constant for price communication;
N: the length of the estimation window;
K: from the utility function $K log(x)$;
$intInterval$: integration period.

Fig. 3. pseudo code of the source algorithm

We also explored an alternative implementation that avoids the complex calculation of the exponential function: from equations (15-16), we have

$$\dot{x}_i = \frac{\beta_i K_i}{\tau_i} - x_i \left(\frac{\beta_i q_i}{\tau_i} + \frac{\alpha_i}{M_i \tau_i} \dot{q}_i \right). \quad (22)$$

This means we could perform the following window update equations

$$\theta = 1 - \left(\beta_i T_s + \frac{\alpha_i}{M_i} \right) \frac{q(k)}{\tau_i} + \frac{\alpha_i}{M_i} \frac{q(k-1)}{\tau_i},$$
$$w(k+1) = \theta w(k) + \beta_i K_i T_s,$$

where no exponential is required.

A simple discretization of the link law (5) is given in the pseudocode of Figure 4. Note the relatively mild computational burden on the routers.

Every packet enqueue:
    $pktCounter ++.$

Every $updInterval$ seconds:

$$aveInput \leftarrow \frac{pktCounter}{updInterval};$$

$$price \leftarrow price + \left(\frac{aveInput}{virtCap} - 1 \right) * updInterval;$$

$$prob \leftarrow 1 - \Phi^{-price};$$

$$pktCounter \leftarrow 0.$$

Every packet dequeue:

    $temp \leftarrow$ uniform();
    if $temp \geq prob$ marking the packet .

Variables:
$price, \quad prob, \quad pktCounter$.

Parameters:
Φ: constant, the same as that of the sources;
$virtCap$: Virtual Capacity, $\gamma * Capacity$;
$updInterval$: update period.

Fig. 4. pseudo code of the link algorithm

V. SIMULATION RESULTS

In this section, we simulate the algorithm with ns-2 to validate the desired performance of the new protocol, including the steady-state properties such as empty queue, fairness, high utilization, and dynamics. We also compare it with TCP-NewReno sources combined with other Active Queue Management schemes including Newreno/RED, Newreno/AVQ and Newreno/PI.

Over all the simulations, we take *one* set of parameters for each of the protocol pairs. All AQM schemes have ECN marking. We set the constant packet size as 1000 bytes. All the queues in the plots are in *packets*, and rates in *packets/sec*. The utilization of the link is averaged over 20ms.

For the new protocol, which we label "FAST", we set the target link utilization to be $c_{0l}/c_l = 0.95$, stability parameter $\alpha = 0.37$, the utility function $50 \log(x)$, and other parameters

as following: $M_i = 1, N = 31, \phi = 1000, \eta = 0.64, T_s = 10ms, \tilde{T}_s = 5ms$. To validate the performance around equilibrium, we use large buffers.

We set RED parameters by default, i.e., the parameters are configured automatically according to the bandwidth and link delays and $gentle\_$ is set to 1 [9]. We also use the default parameter of PI [6] such as $q\_ref = 50$. For AVQ [18], we use the default parameters except that the expected utilization $\gamma = 0.95$. In the case of two-way traffic, all the $qib\_$ parameters are set to be $true$. The buffer of the bottleneck link is set as the bandwidth-delay product when these AVQ schemes are running.

A. Equilibrium properties with two-way long-lived traffic

Similar to the setup in [15], we first show the steady-state performance of the new protocol in comparison with other TCP/AMQ schemes. All the simulations in this part use two-way long-lived traffic on a single bottleneck link, i.e., the same number of homogeneous flows go through in the backward direction which results in ACK compression.

In Figure 5, the bottleneck link is shared by 50 persistent FTP flows in each direction with the same round-trip propagation delay of 50ms. When the capacity varies from 100Mbps to 8Gbps, the utilization of the link based on AIMD schemes decreases significantly. However, the FAST protocol always keeps high utilization as expected and very small queues.
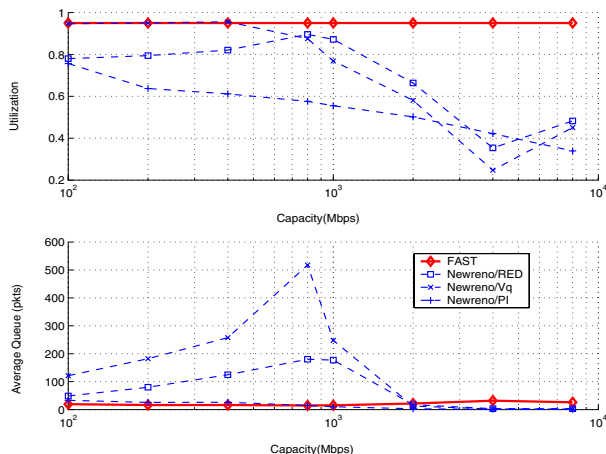

Fig. 5. Impact of Link Capacity

In Figure 6 we leave the capacity fixed at 100Mbps, and the number of flows at 50, but vary the propagation delay from 20ms to 1000ms. Performance of TCP-Reno with all the AQM schemes deteriorate. The new protocol works as desired, keeping high utilization and small queues, except at the point when the RTT is 20ms. In that point with small cwnd, the "sliding window" estimation scheme does not behave well, consistently with the analysis of (19); in future work we will look at improvements to this issue.

What if the number of the flows varies? As indicated in [6], [21], the AIMD schemes tend to be unstable when the number of the flows decrease. In Figure 7, the number of the flows vary in [5 500] with capacity of 1Gbps and round-trip propagation delays of 50ms. Similar as the prior simulations, the TCP schemes show bad utilization when there exists few flows, and a tradeoff
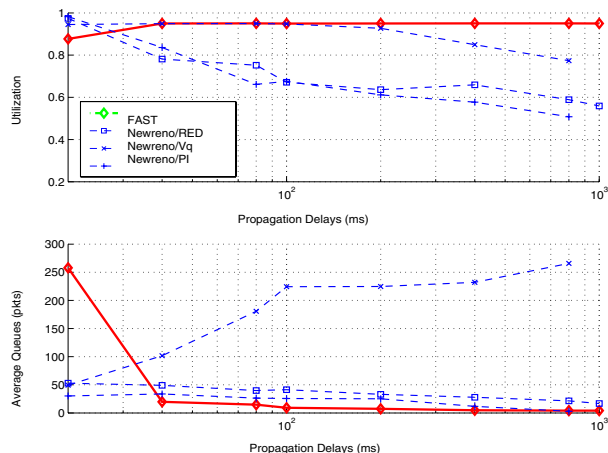

Fig. 6. Impact of Propagation Delays

between utilization and average queues when the utilization is high. In contrast, our protocol still has the desired performance.
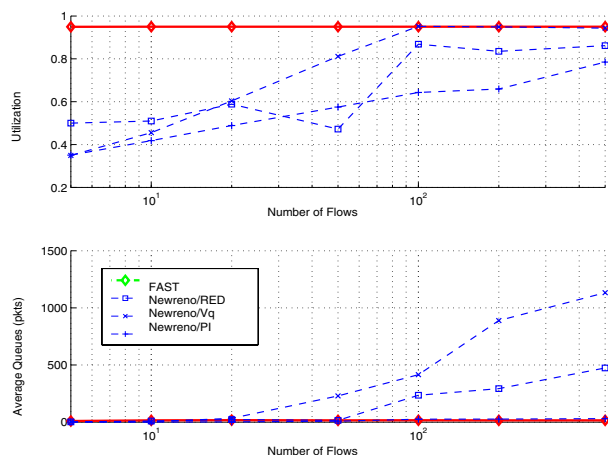

Fig. 7. Impact of Number of Flows

In summary, the AIMD based schemes can achieve high utilization when the equilibrium window is small, as in cases of small bandwidth-delay product or many flows. The new protocol significantly outperforms those schemes when the product is large, i.e., when the sources have to maintain a large equilibrium window. We emphasize that our protocol's parameters were held constant *throughout* this wide range of scenarios.

B. Dynamic Performance with long-lived traffic

Simulations in this section show that the new protocol has fast and stable transient dynamics when *heterogeneous* flows exist and sudden changes in traffic demands take place. One-way long-lived traffic goes through a single bottleneck link with capacity of 2Gbps (250pkts/ms with mean packet size 1000bytes). It is shared at most by 512 ftp flows. The number of flows is doubled every 40 seconds, from 32, to 64, 128, 256, and finally to 512 flows. These groups of flows have round trip propagation delays of 40ms, 80ms, 120ms, 160ms and 200ms respectively. This scenario is designed to stress a high-capacity link with heterogeneous flows.
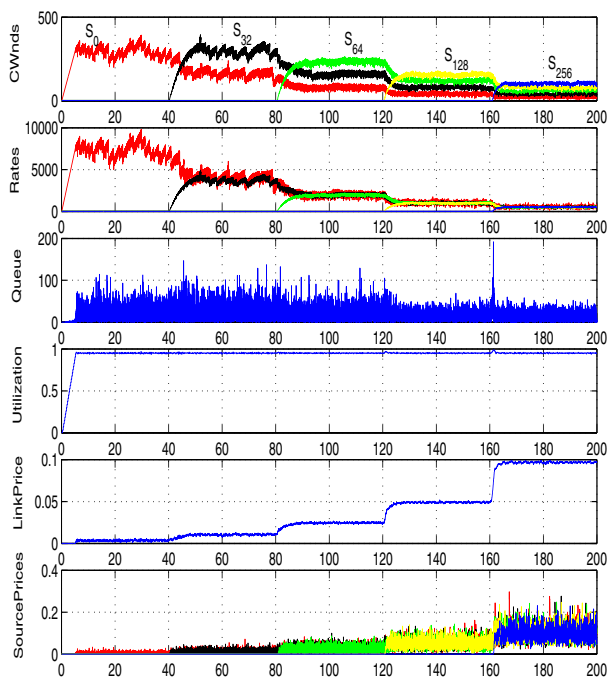
Fig. 8. Dynamic Performance of the New Protocol

In reference to the results of Figure 8, we note:

1. The source rates and the link prices (marking probability) track the expected equilibria when new sources activate. While there is noise in the price estimation, its impact on the windows and rates is moderate, except in the very uncongested case.

2. Fairness is achieved: at each equilibrium stage, the bandwidth is shared equally among sources despite their heterogeneous delays.

3. The queue is small (around 100 packets, less than 0.5 ms of queueing delay) almost all the time, both in transient and in equilibrium. The only (mild) queue overshoot is caused by the activation of 256 new flows in a short time.

4. Link utilization is always around the 95% target even when the traffic demand changes suddenly.

Note that we are not using any "slow-start" phase in this protocol, we are running exclusively the algorithm described before. In fact, at the beginning of the simulation, when the price is small, the sources' rate grows approximately linearly, a fact that can be explained by looking at equation (22). The slope of increase is approximately $\beta_i K_i / \tau_i$, so the utility function's parameter can be used to tune how aggressively the sources start-up, trading off speed with the risk of queue overshoots. If we wished an exponential increase in this initial stage, it may be advantageous to retain a slow-start phase, or use a different utility function, a factor we will explore in future work.

Figure 9 shows the corresponding simulation of the same scenario under Newreno/RED. Features of the trajectory include unstable oscillations of the queue, of significant enough magnitude to deteriorate the utilization of the link. This deterioration is not as severe as in the case of two-way traffic shown before, but is still appreciable. We could attempt to improve on it by tuning of the RED parameters, but this only happens at the expense of higher queueing delays. Similar results (not shown here) are

observed for the other AQMs: they have significant oscillations and are forced to trade off between utilization and delay.
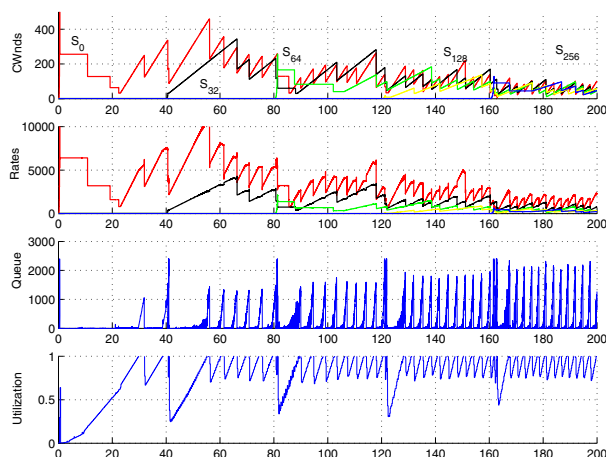

Fig. 9. Dynamic Performance of NewReno/RED

C. Performance with heavy-tailed traffic

The above simulations involve only long-lived traffic, where all sources stay long enough to be controlled. In a real network, flows have been found to follow a heavy-tailed distribution, with short "mice" making up most of the flows, but long controlled "elephants" accounting for most of the packets. How does the new protocol behave in this environment?

We experiment with 512 sources with heterogeneous delays as in the previous section (from 40ms to 200ms, only some of which are plotted in the figure below) on a link of 1Gps. Flow sizes are randomly generated according to a Pareto distribution, where more than 90% of the flows have sizes less than 1000 packets, but only contribute around 25% of the overall packets. All sources generate such flows with inter-arrival times exponentially distributed.

The new protocol still keeps high utilization and small queue as in the long-connection cases, see Figure 10. There exist only a few queue overshoots, and the elephants share the bandwidth fairly.

VI. AN IMPLEMENTATION BASED ON QUEUING DELAY

The protocol described above achieves close to full utilization of link capacity, while at the same time operating under essentially empty queues. These two properties can only be simultaneously achieved by some form of explicit congestion notification. In the absence of it, congestion can only be detected through some of its undesirable effects, such as the appearance of queueing delays. However, if one could guarantee that these delays are moderate, it would perhaps be preferable to avoid the burden of ECN signaling on the routers.

In this regard, the fact that the prices in our protocol are virtual queueing delays, suggests the possibility of using *real* queuing delays as a price signal; this amounts to choosing $c_{0l} = c_l$ in the link equation (5). The sum of such delays over congested links can be estimated by sources by subtracting the minimum observed RTT (taken to be propagation delay) from the current
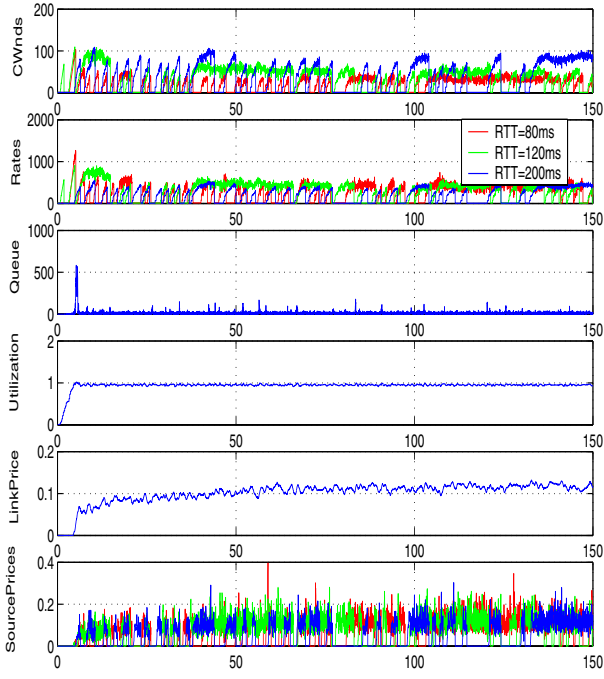
Fig. 10. new protocol with heavy-tailed traffic

RTT, avoiding any explicit signaling between links and sources. This is precisely the type of congestion feedback used in TCP-Vegas [2]. The question before us is to find a source protocol that can achieve (i) small equilibrium prices (delays), (ii) freedom of choice in resource allocation, and (iii) stability, working with queueing delay as a price signal.

A. A modified control law

The appearance of queueing delays around equilibrium would invalidate the theoretical analysis of Section III; indeed, in analyzing equations (15-16) around equilibrium, we interpreted the quantity τ_i to refer to the fixed (propagation/processing) delay d_i. This allowed us to treat it essentially as a parameter when performing the linearization to the form (13). Here, instead, τ_i will be the *variable* quantity

$$\tau_i = d_i + q_i, \qquad (23)$$

where q_i is the queueing delay observed by the source. Since q_i is also the price, the dynamics must be modified if we wish a linearization (13); below is the proposed alternative:

$$\dot{\xi}_i = \frac{\beta_i}{(d_i + q_i)}(U_i'(x_i) - q_i), \qquad (24)$$

$$x_i = x_{m,i} e^{\xi_i} \left(\frac{d_i}{d_i + q_i}\right)^{\frac{\alpha_i}{M_i}}. \qquad (25)$$

Here, (24) is unchanged from (15), we have only made explicit the relationship (23) for the RTT. In particular it imposes the equilibrium relationship $U_i'(x_{i0}) = q_{i0}$. and its linearization (dropping the subscript i) is

$$\delta\dot{\xi} = \frac{\beta}{(d + q_0)}\big(U''(x_0)\delta x - \delta q\big) = \frac{\beta}{\tau_0}\big(-\frac{\delta x}{\nu} - \delta q\big). \quad (26)$$

This is identical to (17) where τ_0 is now the equilibrium RTT. Note that the q in the denominator of (24) does not contribute to the linearization since the multiplying factor is zero at equilibrium. Equation (25) involves a more substantial change from (16), but has been designed to give the same linearization in the current circumstances. Indeed we have

$$\delta x = x_0 \delta\xi - x_m e^{\xi_0} \frac{\alpha}{M} \frac{d^{\frac{\alpha}{M}}}{(d + q_0)^{\frac{\alpha}{M}+1}}\delta q$$

$$= x_0 \left(\delta\xi - \frac{\alpha}{M\tau_0}\delta q\right),$$

coinciding with (18). From here on, the same analysis as in Section III-B leads to the linearized source dynamics (13).

Does this mean that our local stability theorem would hold for this protocol? Unfortunately there is another difficulty that arises from queueing delays; namely, that the network equations (1) and (2) become time-varying due to variations in the delays $\tau_i(t)$. This means that the analysis by Laplace transforms is at best an approximation, obtained by considering variations only in the *dependent* variables, but not in the *independent* variable. This kind of approximation has been used successfully [5], [21] to analyze TCP-Reno dynamics, but has not been rigorously justified. Thus there is hope that this protocol would behave satisfactorily, but we must rely (more so than before) on empirical simulations to validate this fact.

B. Implementation and Simulations

Similarly to Section IV, we can discretize (24-25) with time interval T_s, as an implementation for the source dynamics; then we have the following congestion window update:

$$\xi_i(k) = \xi_i(k-1) + \beta_i \left(\frac{K_i}{w_i(k-1)} - \frac{q_i(k)}{d_i + q_i(k)}\right) T_s, \text{(27)}$$

$$w_i(k) = w_{m,i} e^{\xi_i(k)} \left(\frac{d_i}{d_i + q_i(k)}\right)^{\frac{\alpha_i}{M_i}}, \qquad (28)$$

Other than the slightly different equations, the main change in the algorithm is that the price $q_i(k)$ is now obtained from the estimation of queueing delay, as described before. At the link side, a router with no AQM has exactly the same dynamics as (5), except scaled by the real capacity, provided the buffer is large enough to accommodate the required queueing delays.

For concreteness, in the simulations below we again use the utility function $U_i(x) = K_i \log(x_i)$; this case is even more similar to TCP Vegas, which corresponds to this kind of utility function (see [22]); however Vegas may be unstable (see [3]). Our dynamics provides a way to achieve the same equilibrium in a stable way. For another strategy to stabilize Vegas, see [3].

Figure 11 uses the same scenario and required parameters values as in Section V-B. The simulation shows fast response to the traffic demand and stability. Furthermore, the windows are extremely smooth as well as the queues due to the accurate estimation of the price, i.e., the queueing delay. This, and the lack of complexity at routers, are interesting advantages of this protocol with respect to the ECN version. There are, however, drawbacks: one, that a certain amount of queueing delay must be tolerated here. While parameters (e.g. K_i in the utility function) can be tuned to make it small, there is a tradeoff with the

speed of response of the system. Another issue that appears in the simulation is some unfairness, caused by sources joining the network later that overestimate the propagation delay, and thus underestimate price, taking up a larger share of the bandwidth.
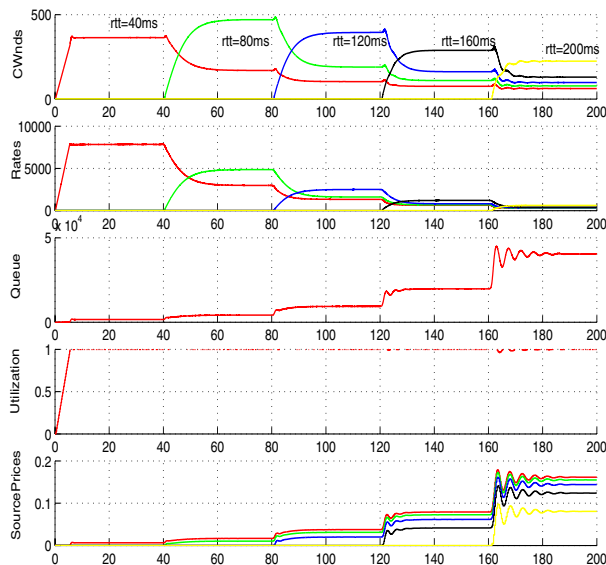

Fig. 11. Dynamics of protocol based on queueing delay

VII. Conclusion

An ideal TCP congestion avoidance method that can achieve high utilization, small queueing delay, freedom from oscillations and fairness in bandwidth allocation has been a major objective of networking research in recent years. Our results show first that at the level of fluid-flow quantities, these objectives are indeed achievable if one assumes a scalar price signal that can be fed back to sources from links. We have further demonstrated a practical version of the protocol, based on ECN marking, that appears to successfully approximate these objectives in high capacity scenarios where current protocols exhibit limitations, and showed that fixed parameter choices can accommodate a wide range of scenarios. More research is required to verify the generality of these conclusions. We are also working [26] on a better understanding of the nonlinear dynamics, which has significant impact on the speed of the control away from equilibrium.

Another recent proposal to achieve the same goals has been XCP [15]. While this work contains a stability theorem for *homogeneous* delays, their second-order control laws do not allow the argument to extend to heterogenous delays, as is done in this paper. On the performance side, our results of Section V-A are comparable but ours is based on a far simpler implementation with only one ECN bit sent from links to sources, and no information sent from sources to links.

The main obstacle to the implementation of these kinds of protocols is that they require substantial changes to current practice at both routers and end-systems. For this reason we were motivated to consider the implementation based on queueing delay, similar to TCP Vegas, which appears capable of delivering most of the benefits with no active participation of network routers. Based on our preliminary success in simulations, we

are currently pursuing experimental deployment of these kinds of protocols [4]. Part of this effort will involving testing the coexistence of this protocol with the existing ones.

References

[1] S. Athuraliya, V. H. Li, S. H. Low, and Q. Yin, REM: active queue management, , *IEEE Network*, vol. 15, no. 3, pp.48-53, 2001.
[2] L. S. Brakmo and L. Peterson, TCP Vegas: end to end congestion avoidance on a global Internet, *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 8, October 1995.
[3] D. H. Choe and S. H. Low, Stabilized Vegas. *to appear in 2003 IEEE Infocom*, http://netlab.caltech.edu/
[4] FAST Project, http://netlab.caltech.edu/FAST/.
[5] C. Hollot, V. Misra, D. Towsley, and W-B Gong, A control theoretic analysis of RED, in *Proc. IEEE Infocom*, April 2001.
[6] C. Hollot, V. Misra, D. Towsley, and W-B Gong, On designing improved controllers for AQM routers supporting TCP flows in *Proc. IEEE Infocom*, April 2001.
[7] V. Firoiu and M. Borden, A study of active queue management for congestion control, in *Proc. IEEE Infocom*, March 2000.
[8] S. Floyd and V. Jacobson, Random early detection gateways for congestion avoidance, *IEEE/ACM Trans. on Networking*, vol. 1, no. 4, pp. 397-413, August 1993.
[9] S. Floyd, RED: Discussions of Setting Parameters, http://www.icir.org/floyd/red.html#parameters.
[10] S. Floyd, R. Gummadi, and S. Shenker. "Adaptive RED: An Algorithm for Increasing the Robustness of RED's Active Queue Management", Preprint, available at http://www.icir.org/floyd/papers.html, August 2001.
[11] S. Floyd, "HighSpeed TCP for large congestion windows", Internet draft, June 2002. http://www.ietf.org/internet-drafts/draft-floyd-tcp-highspeed-00.txt
[12] R.J. Gibbens and F.P.Kelly, Resource pricing and the evolution of congestion control, *Automatica* 35 (1999), 1969-1985.
[13] V. Jacobson, Congestion avoidance and control, *Proc. SIGCOMM'88, ACM*.
[14] R. Johari and D. Tan, End-to-end congestion control for the Internet: delays and stability, *IEEE/ACM Transactions on Networking* 9(2001) 818-832.
[15] D. Katabi, M. Handley, and C. Rohrs, "Internet Congestion Control for High Bandwidth-Delay Product Networks." *Proc. ACM Sigcomm*, Pittsburgh, PA, August 2002.
[16] F. P. Kelly, A. Maulloo, and D. Tan, Rate control for communication networks: Shadow prices, proportional fairness and stability, *Jour. Oper. Res. Society*, vol. 49, no.3, pp 237-252, March 1998.
[17] F.P. Kelly, Models for a self-managed Internet, *Philosophical Transactions of the Royal Soceity*, 358(2000) 2335-2348.
[18] S. Kunniyur and R. Srikant, "Analysis and Design of an Adaptive Virtual Queue (AVQ) Algorithm for Active Queue Management", *Proc. ACM Sigcomm*, San Diego, Aug 2001.
[19] S. Kunniyur and R. Srikant, A time-scale decomposition approach to adaptive ECN marking, *IEEE Trans. on Automatic Control,* Vol. 47, No6., pp. 882-894, June 2002.
[20] S. H. Low and D. E. Lapsley, Optimization flow control, I: basic algorithm and convergence, *IEEE/ACM Transactions on Networking*, vol.7, no.6,pp861-874, December 1999.
[21] S. H. Low, F. Paganini, J. Wang, S. A. Adlakha, and J. C. Doyle, Dynamics of TCP/RED and a scalable control, Proc. IEEE Infocom, June 2002.
[22] S. H. Low, F. Paganini, J. C. Doyle, Internet congestion control, *IEEE Control Systems Magazine*, February 2002.
[23] F. Paganini, J. C. Doyle,and Steven H. Low, Scalable laws for stable network congestion control, in *Proc. IEEE Conference on Decision & Control*, Orlando, FL, 2001.
[24] G.Vinnicombe, On the stability of end-to-end congestion control for the Internet, Tech. Rep., Cambridge University, CUED/F-INFENG/TR.398, December 2000.
[25] G. Vinnicombe, On the stability of networks operating TCP-like congestion control, Proc. IFAC'02. http://www-control.eng.cam.ac.uk/gv/internet/index.html.
[26] Z. Wang and F. Paganini, Global Stability with Time Delay in Network Congestion Control, *Proc. IEEE Conf. on Decision and Control,*, Las Vegas, NV, 2002.